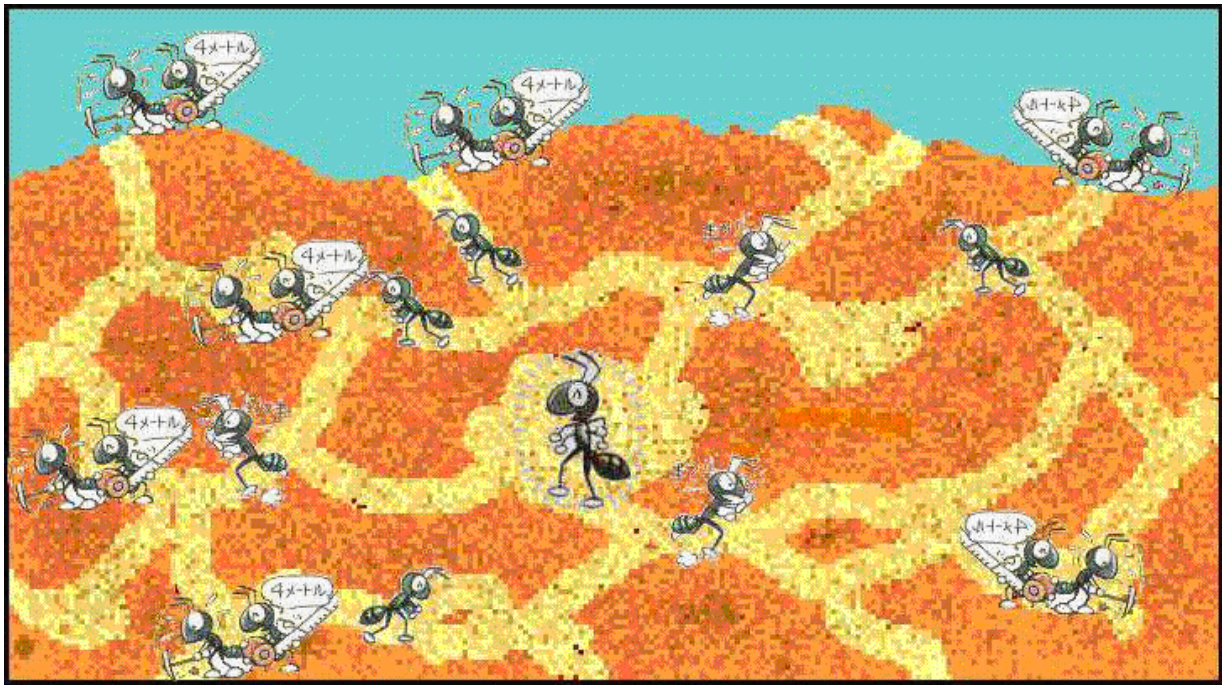
Reactive Sensor Networks

Richard R. Brooks
Head
Distributed Intelligent Systems Dept.
Applied Research Laboratory
Pennsylvania State University
P.O. Box 30
State College, PA 16804-0030
email: rrb@acm.org
Tel. (814) 863-5698
Fax (814) 863-1396
Dept. (814) 863-5735



Acknowledgment / Disclaimer

This effort is sponsored by the Defense Advance Research Projects Agency (DARPA) and the Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-99-2-0520 (Reactive Sensor Network). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the author's and should not be interpreted as necessarily represent the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency (DARPA), the Air Force Research Laboratory, or the U.S. Government.

RSN Goals

Support Sensor Network data aggregation and flexible tasking by applying collaborative signal processing and mobile code technologies.

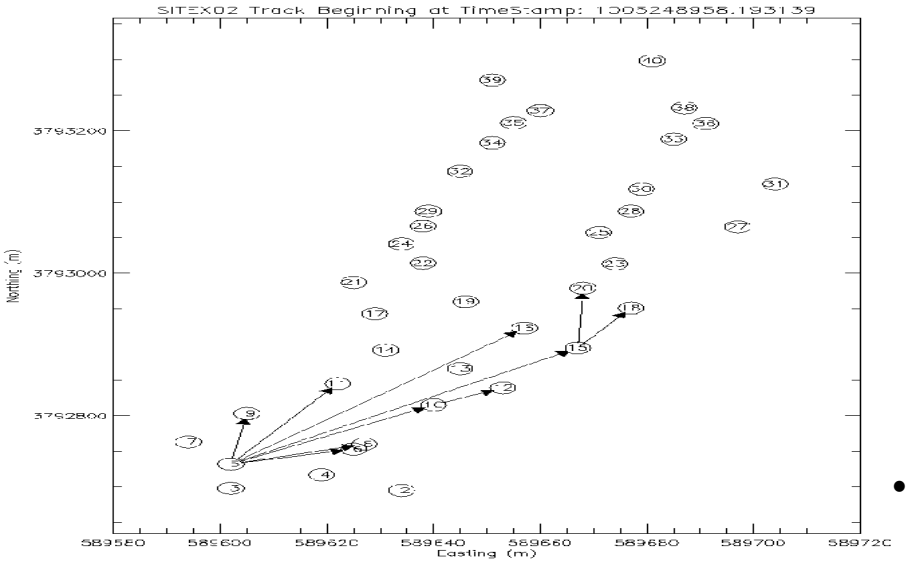
Recent Accomplishments

- Delivery of mobile code daemons for Intel and SH4 Linux
- Creation of target tracking mechanisms
  - Pheromone
  - EKF
  - Bayesian
- Verification of tracking using CA simulations
- Papers accepted for journal special issue:
  - “Self-organized distributed sensor network target tracking”
  - “Traffic model evaluation of ad hoc target tracking algorithms”
- Derivation of target tracking approach for operational demonstration
- Implementation, integration, and test of operational demonstration software. (> 5 projects collaborating).

Operational Demonstration

Applied Research Laboratory

Collaborative Tracking Network:



PSU/ARL RSN Objectives:

- Show decentralized target tracking
- Demonstrate self-organizing sensor network
- Test robustness to node and network disruptions
- Find limitations due to node density
- Filter false positives from system
- Computations adapt to changing network structure

Methodologies used:

- Computations use robust Closest Point of Approach (CPA) statistic
- Diffusion routing limits information propagation
- Data association from local information
- Euclidean metric finds track with best-fit
- Kalman filter smooths velocity estimates
- Modular mobile code framework for

Expected Results:

- **System conserves resources (12/01):**
  - Time series reduced to CPA event
  - False alarms filtered locally
  - Multiple CPA events become one track even
- **Data association implementation (02/02):**
  - Improve the Euclidean metric
  - Integrate certainty values from SitEx
  - Integrate target classification
- **Effect of node laydown (05/02):**
  - Density of nodes vs. target density
  - Density of nodes vs. target maneuvers
- **System performance functions (06/02):**
  - System dependability from node distributi
  - Result accuracy from node distribution

Operational Demonstration

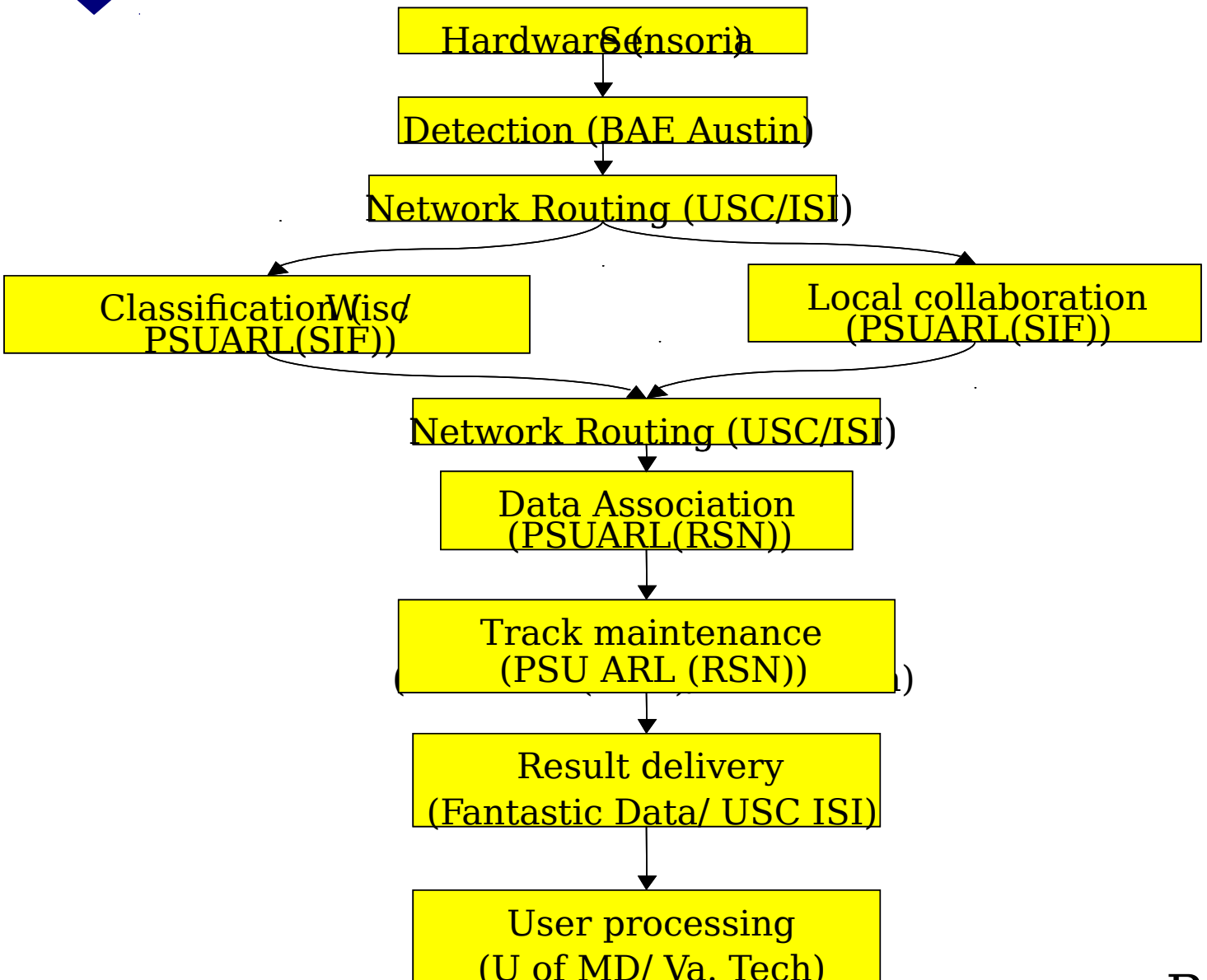
- Autonomous sensor nodes deployed
- Target vehicles traverse sensor field
- “Clumps” of sensors exchange information
- One node in clump estimates target heading, speed, position
- Target parameters used to match existing (create new) tracks
- New parameters merged with existing ones
- Track information reported to user workstation
- Track information propagated in advance of target
- Difficult global problem decomposed into tractable local problems

ColTraNe approach

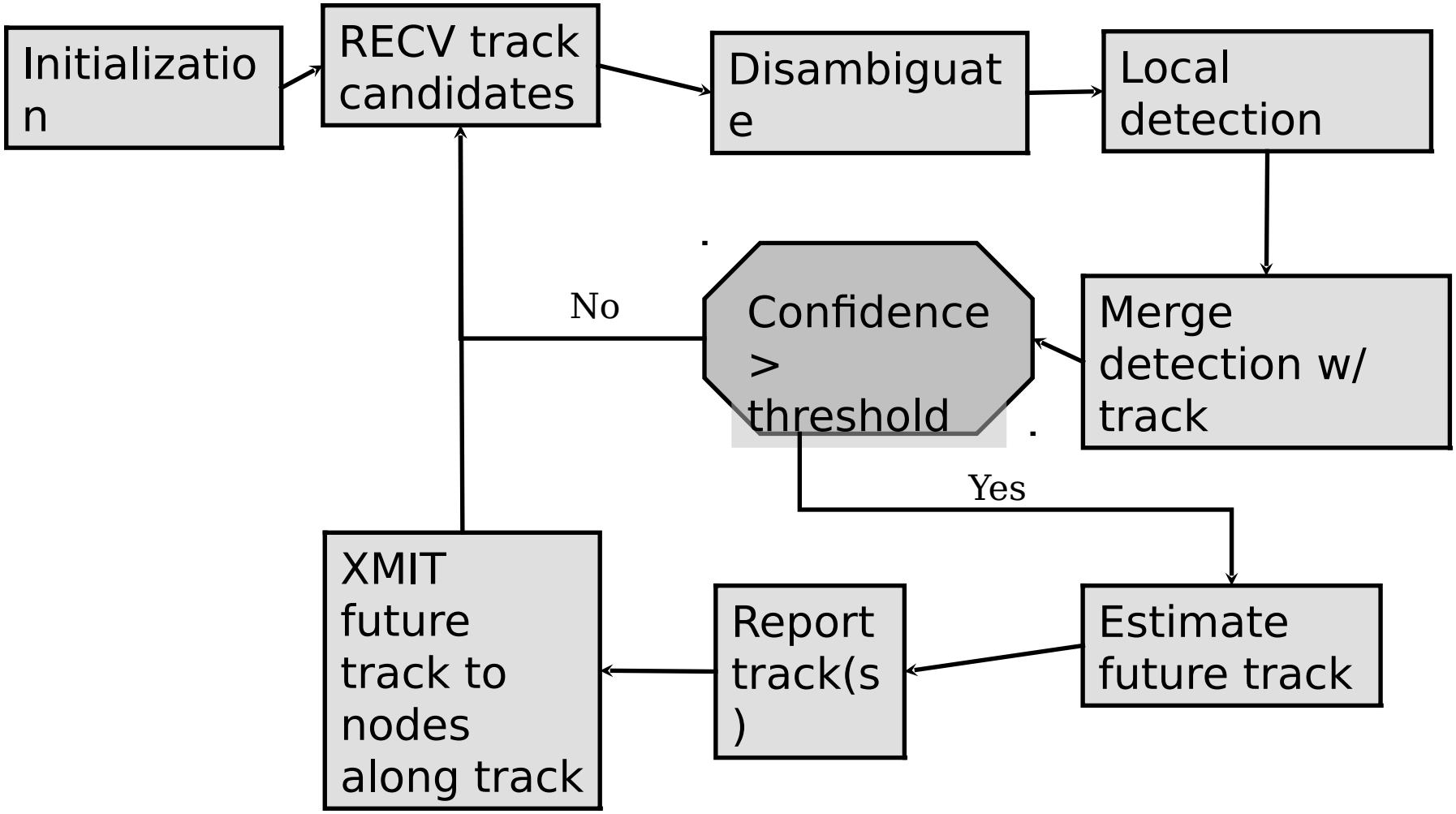
Embed target tracking logic in the network:

- Sub-problems with multiple possible solutions
- Self-organized node tasking built on network routing API
- Local detection, classification, and parameter estimation
- Local data association and ambiguity resolution
- Local prediction of future trajectory

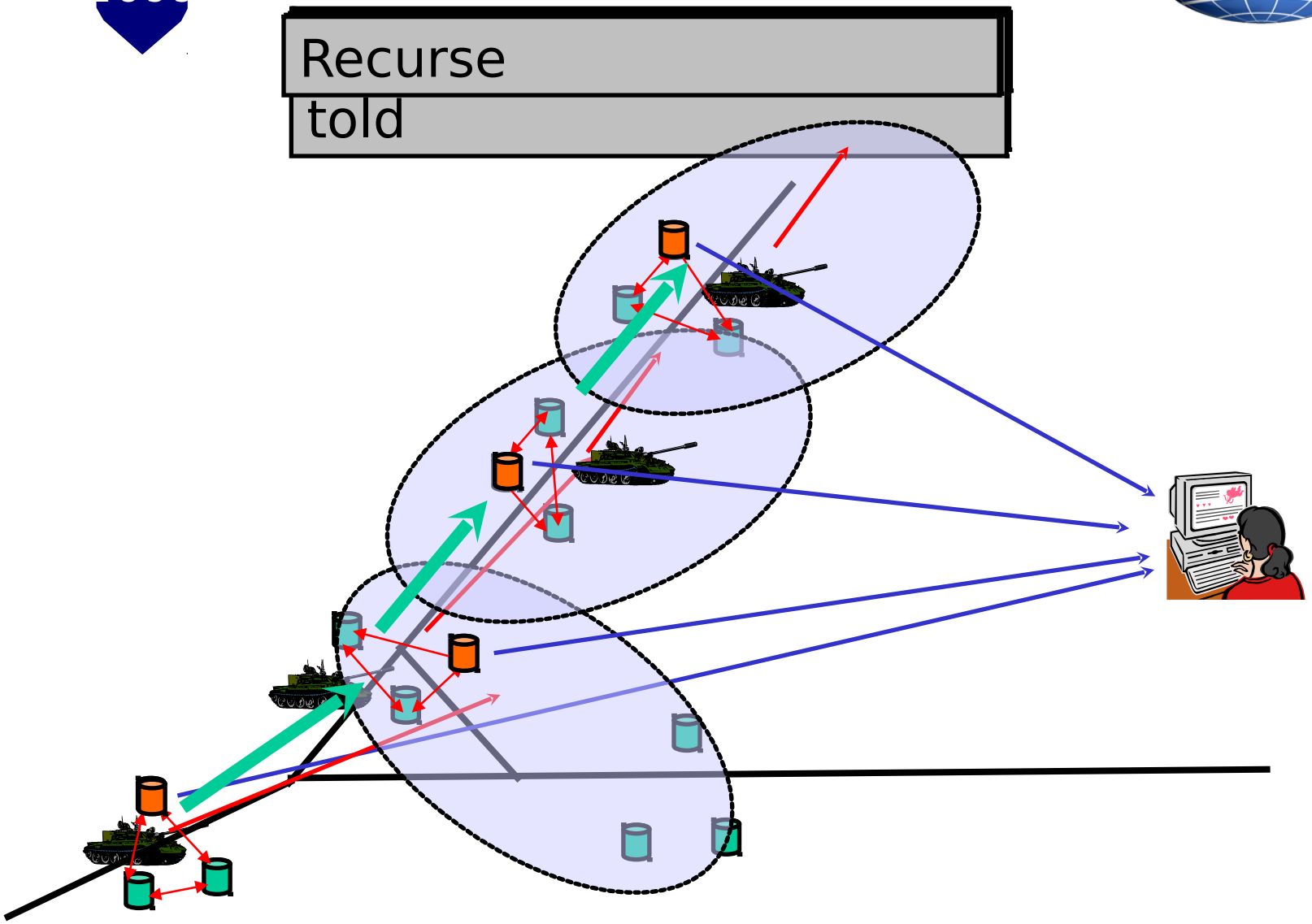
Teams


Hardware (Sensoria)
Detection (BAE Austin)
Network Routing (USC/ISI)
Classification (Wisc/ PSUARL(SIF))
Local collaboration (PSUARL(SIF))
Network Routing (USC/ISI)
Data Association (PSUARL(RSN))
Track maintenance (PSU ARL (RSN))
Result delivery (Fantastic Data/ USC ISI)
User processing (U of MD/ Va. Tech)

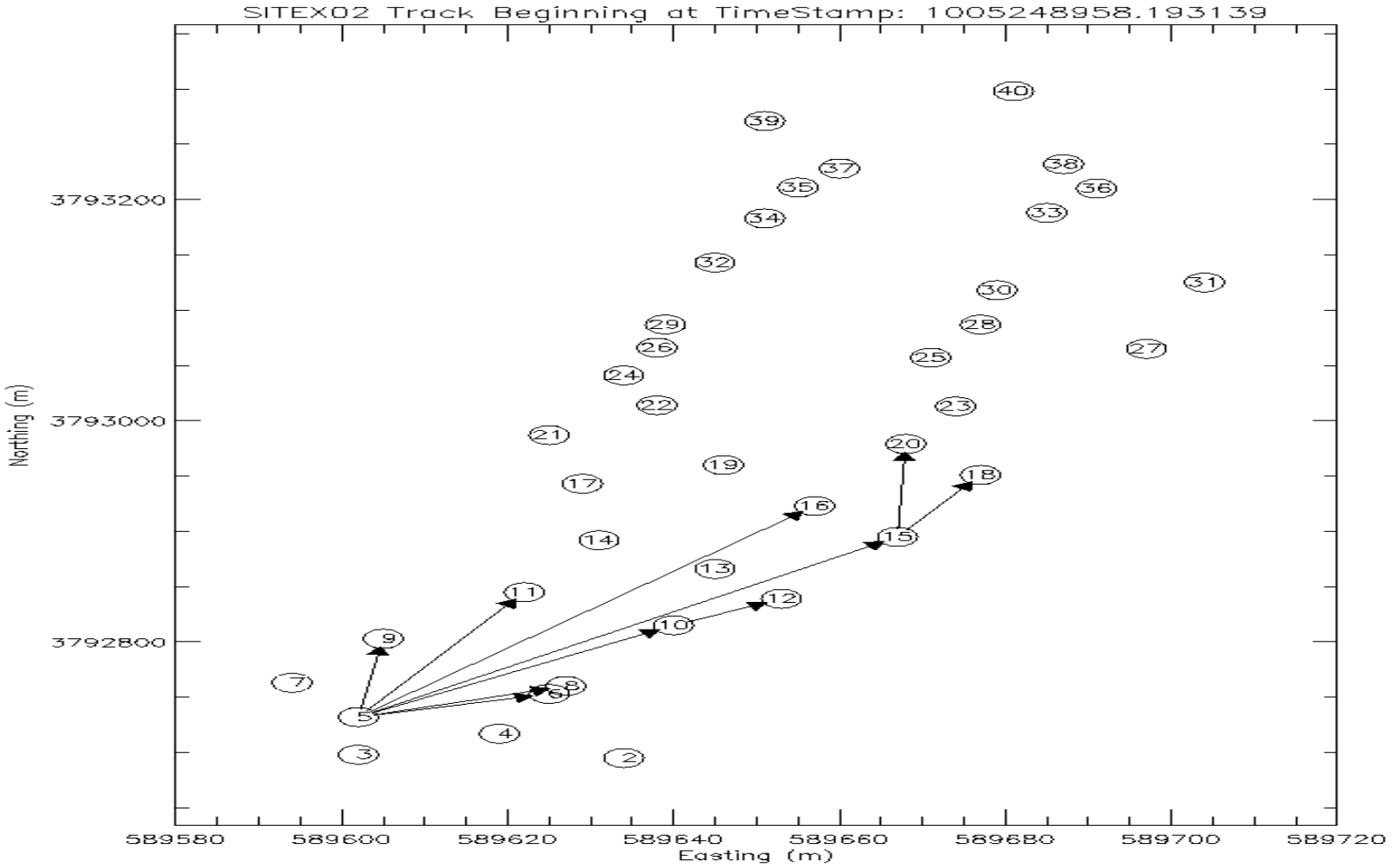
ColTraNE flowchart



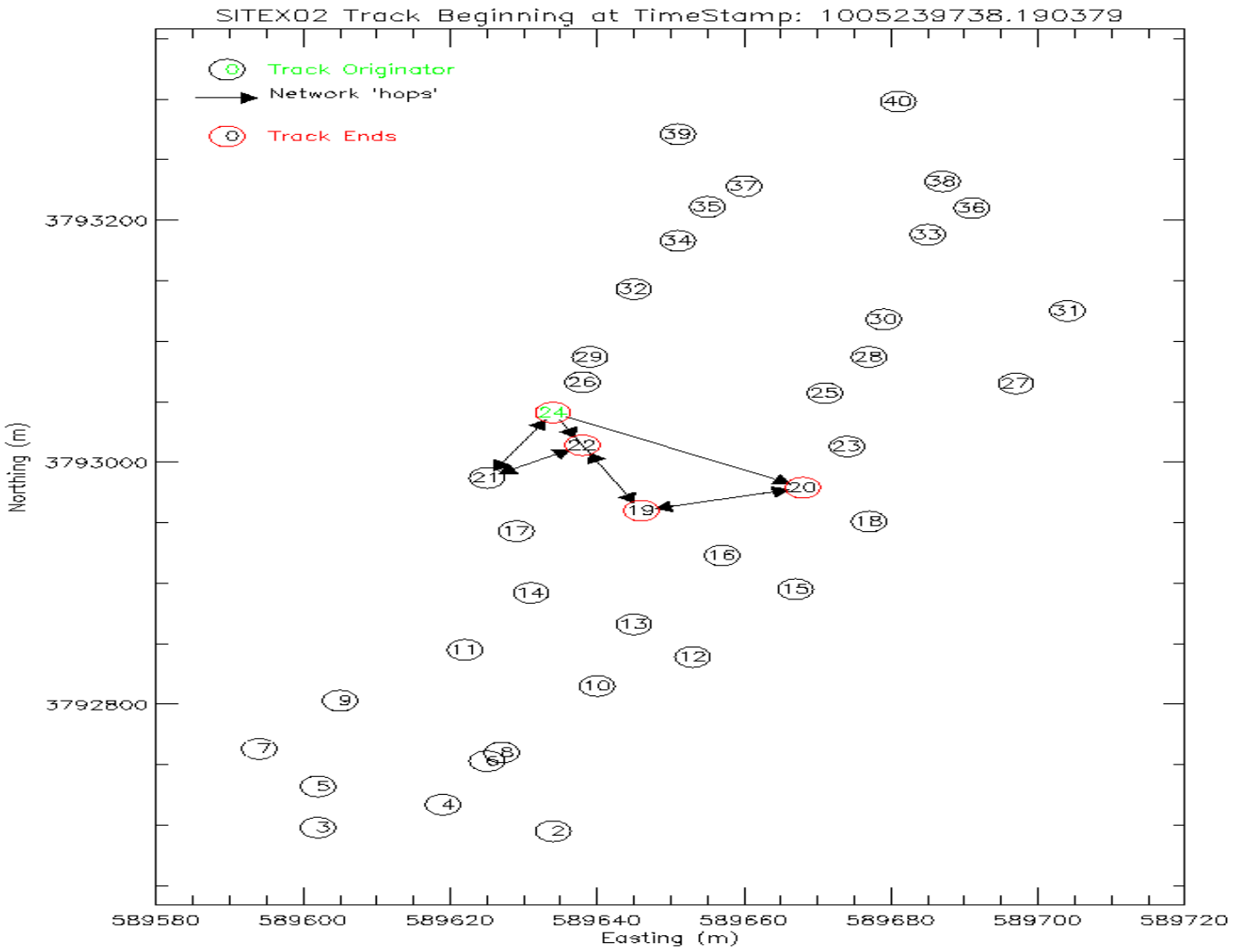
Demonstration scenario



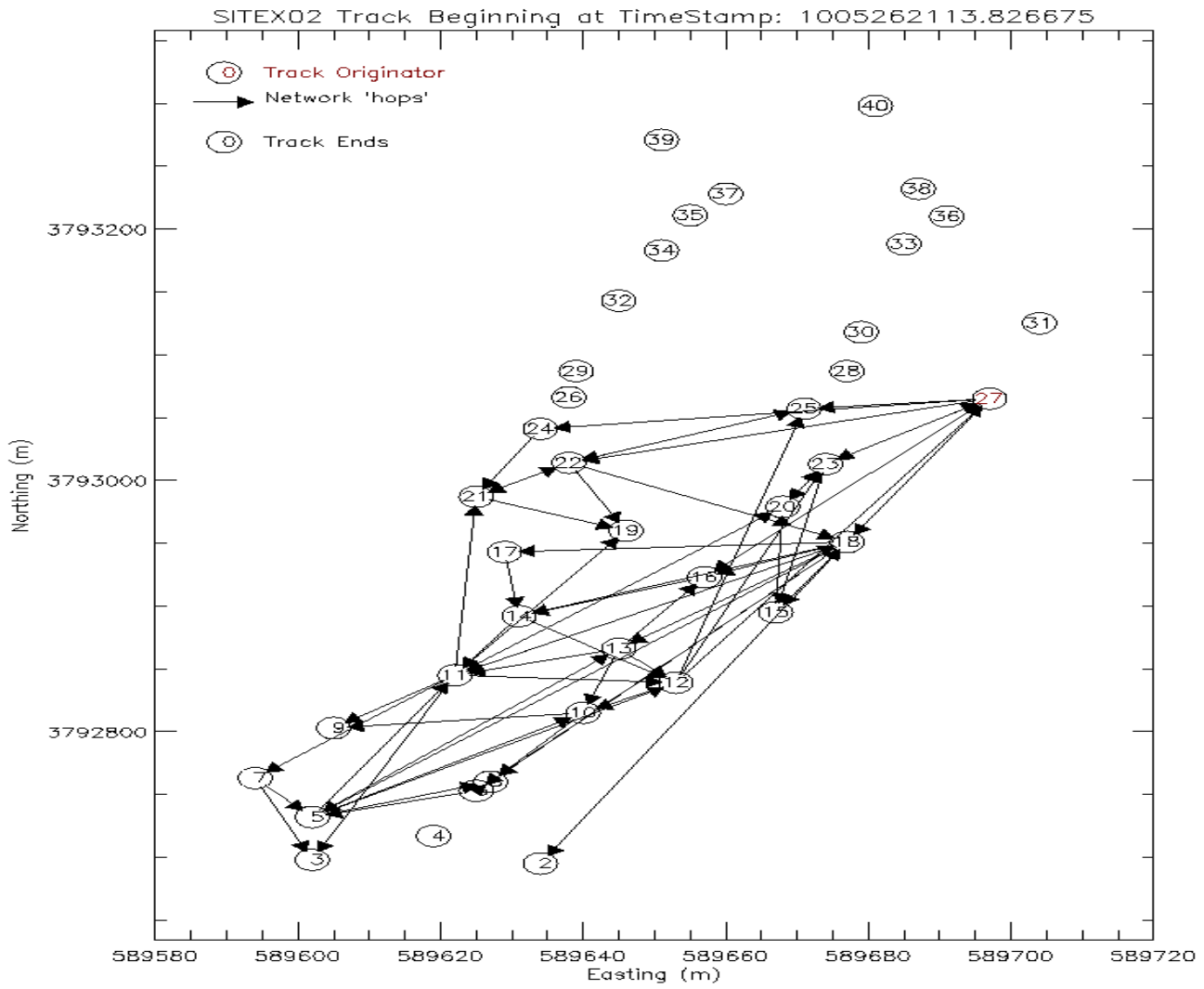
29 Palms tracks – 1 Good



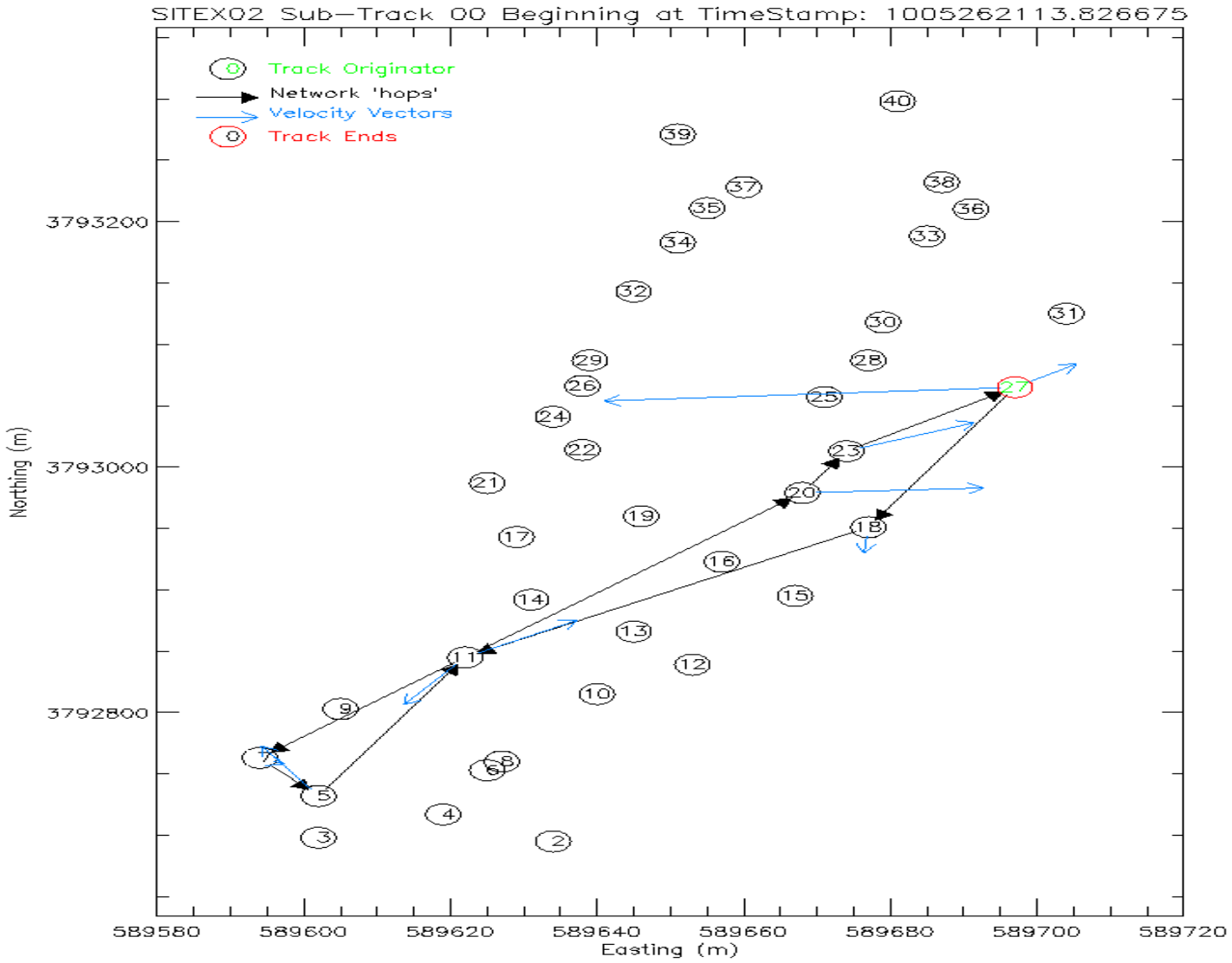
29 Palms tracks – 2 Bad



29 Palms tracks – 3 Ugly



Tracks – Ugly revisited



29 Palms results – 1

- During Sitex 55% of the velocities calculated were 0.0 mph
- 0.0 mph returned by velocity calculation when:
  - Not enough CPA's available for reliable calculation
  - No discernable correlation between CPA events
- 55% of the collaborative detections were false alarms
- When 0.0 returned track information is not propagated
- Higher level processing in the network filters

29 Palms results – 2

- Several matches between clumps with magnitude < 0.0001
- Euclidean metric between predicted at last hop and calculation at current hop less than 0.0001 meter
- Cause – incorrect clumping causing neighbors to form clumps and continue track. Velocity estimation using almost identical data. (Correction underway).
- Note: That means the following:
  - Detection
  - CPA transmission and reception
  - Velocity estimation
  - Track matching
  - Track continuation
  - Track transmission and reception

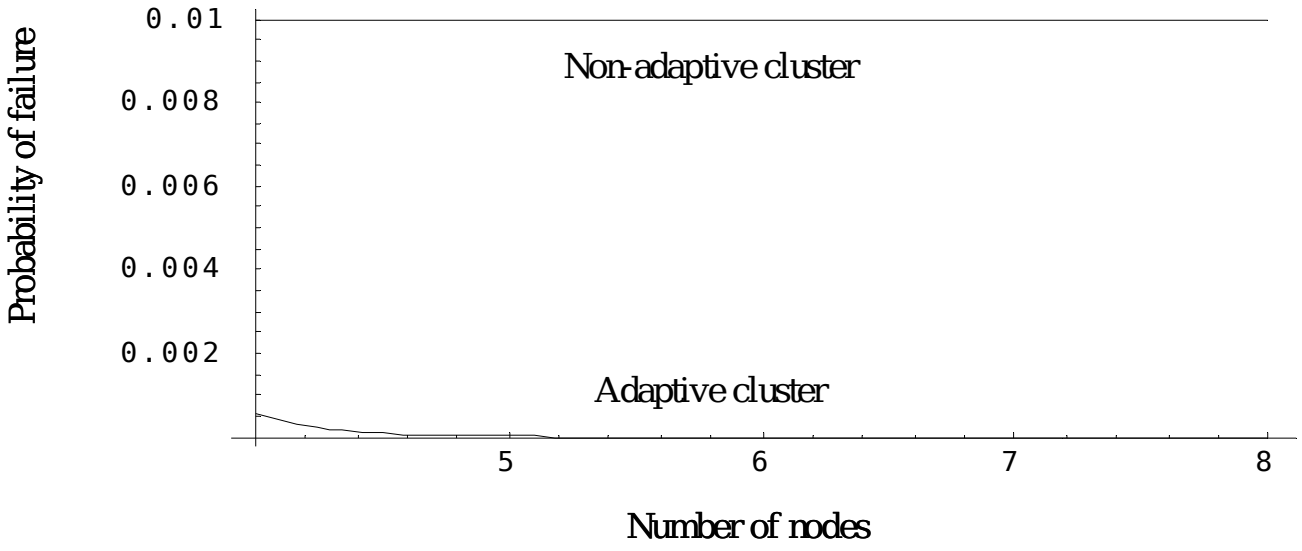
All were completed during the time it took the target to move between adjacent nodes at 29 cm/s

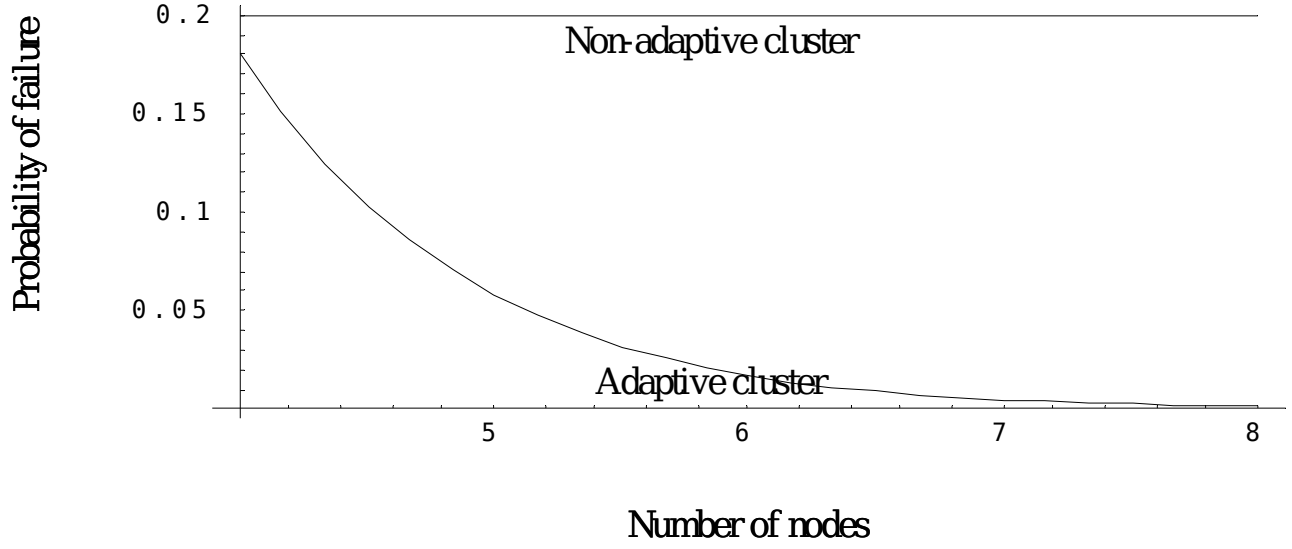
29 Palms results – 3

- In fielded system, classification done using U. of Wisc. or SIF classifier chosen at node boot.
- Next phase – derive a decision function from confusion matrices and current target classes to choose classifier.
- Mobile code daemon automatically downloads, installs, and calls new classifier if necessary.

$$D\left(\begin{array}{c|cccc} & \text{Heavy tracked} & \text{Light tracked} & \text{Heavy wheeled} & \text{Light wheeled} \\ \hline \text{Heavy tracked} & 0.65 & 0.25 & 0.1 & 0 \\ \text{Light tracked} & 0.3 & 0.7 & 0 & 0 \\ \text{Heavy wheeled} & 0 & 0.1 & 0.8 & 0.1 \\ \text{Light wheeled} & 0.05 & 0.32 & 0.10 & 0.53 \end{array}, \begin{array}{c} 4 \\ 1 \\ 0 \\ 3 \end{array}\right)$$

Clump Robustness

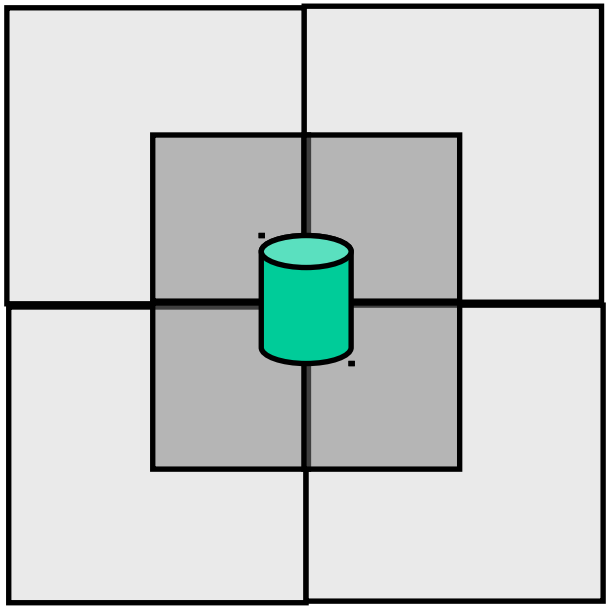




Next steps

- Evaluate versus ground truth
- Re-insert Extended Kalman Filter
- Improve Euclidean metric
  - Add heading discrimination using target turning radii
  - Include velocity uncertainty in heading discrimination
  - Modify thresholds
  - Use classification confusion matrix information
- Use mobile code interfaces for automatic software reconfiguration (ex. classification)
- Distributed tree pruning with lateral inhibition

Self-organization



Relative to its position, each node detects targets and informs other nodes along the predicted trajectory. It produces track hypotheses.

Self-reproducing systems are *Autopoetic* systems [Maturana and Varela], with these characteristics:

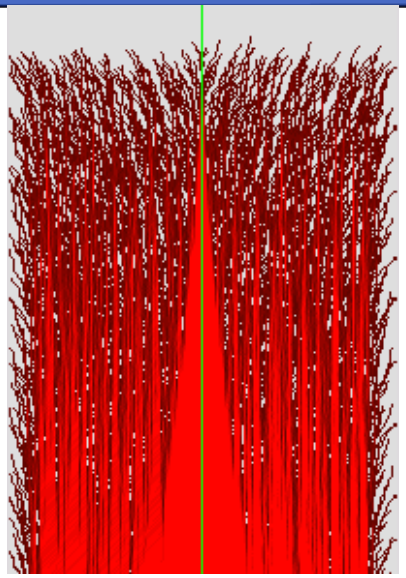
- Self-referential
- Distinguish self from others
- Have internal structure
- Non-closed systems

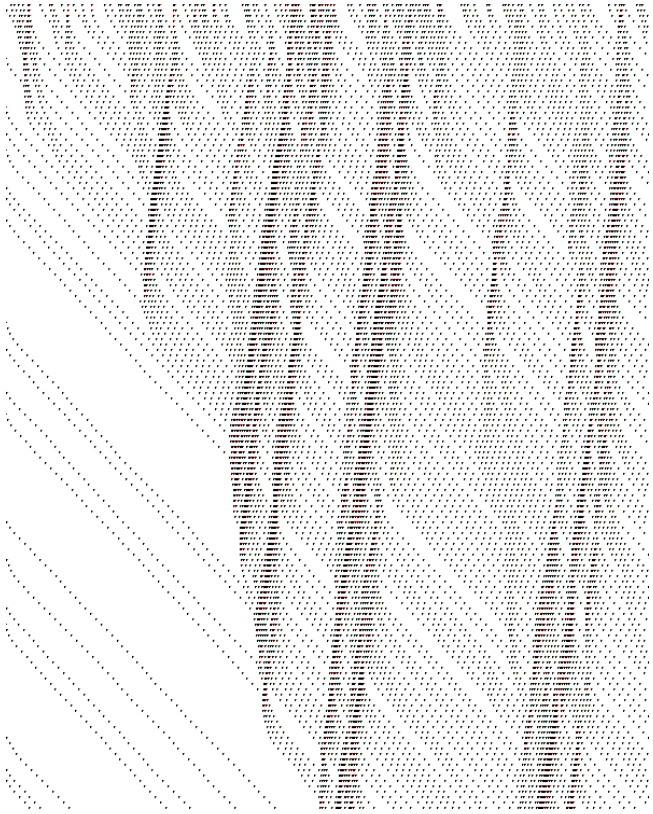
Interacting automata models

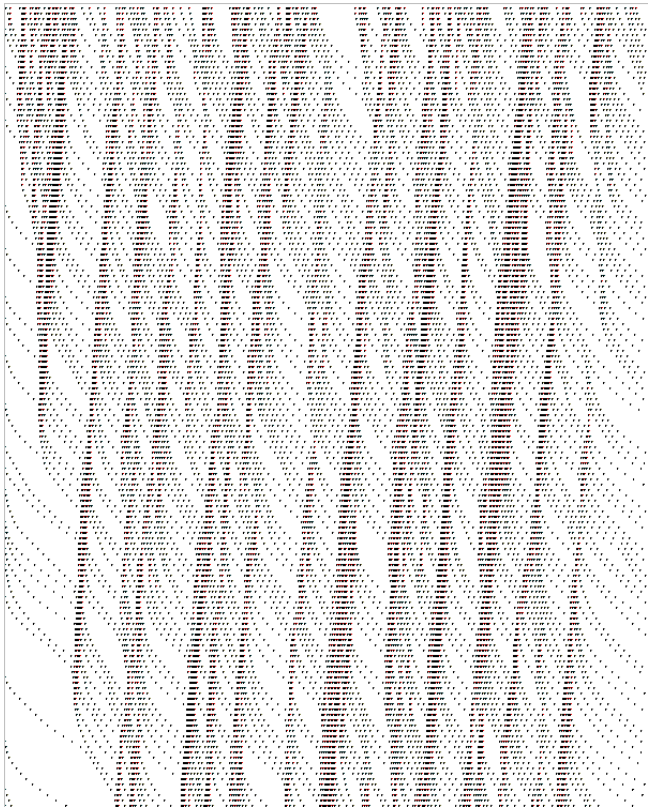
Traffic analysis models pioneered at Los Alamos:

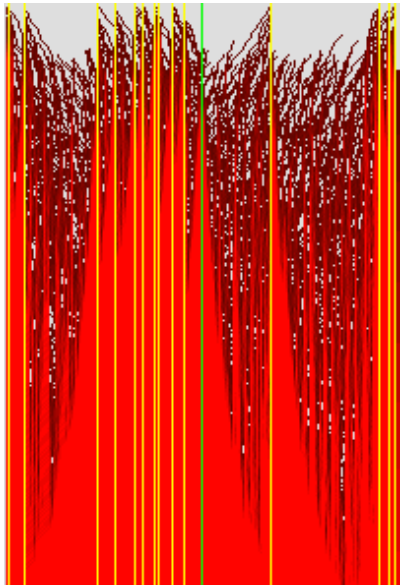
- Used to analyze vehicular traffic
- Nodes represent highway
- Behavior of vehicles influenced by environment
- Allow study of traffic jam behavior / Internet traffic
- We are adding autonomous agent interactions
- Allows study of interactions between tracking algorithms and network behavior

Examples









Intelligent societies

Highly evolved societies of cooperating individuals,

with these characteristics:

- Construction of climate controlled communal housing
- Individuals altruistically sacrifice themselves for the common good
- Equitable distribution of work
- Division of tasks among castes of specialized workers
- Domestication of other species
- Creation of logistic networks to support cities and war efforts

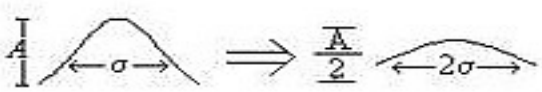
These societies control most of the air and land space on earth

Algorithm 1 - Pheromones

Pheromone interactions

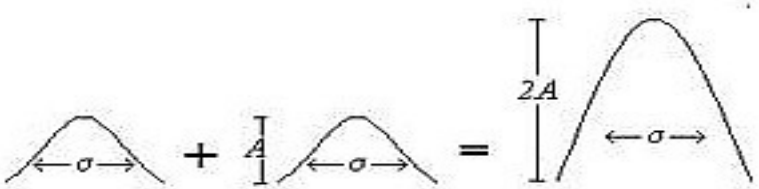
With each time step

A Pheromone signal



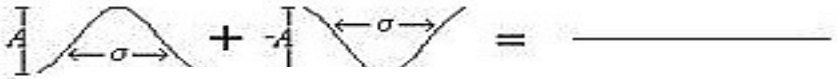
Become less intense and more diffuse

Two positive pheromone signals combined

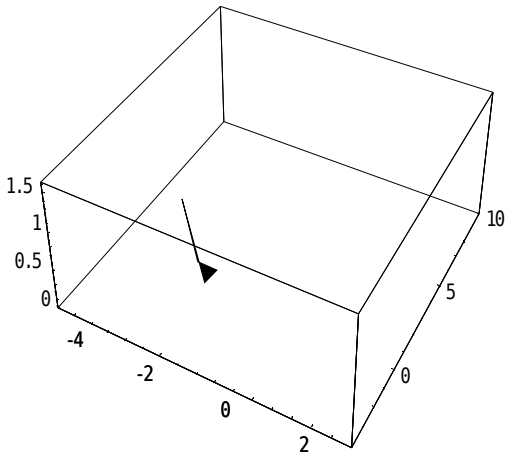


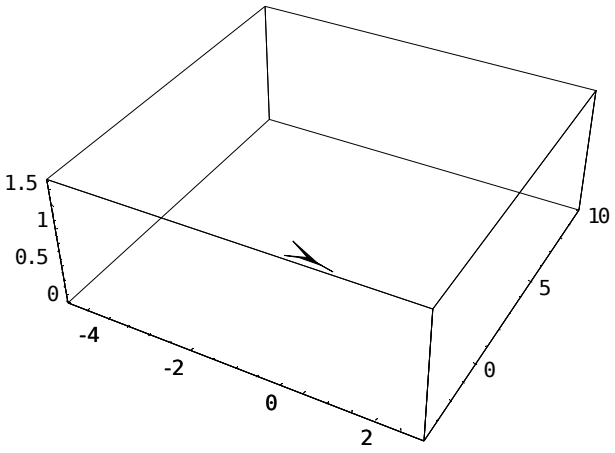
Become a more intense signal of the same type

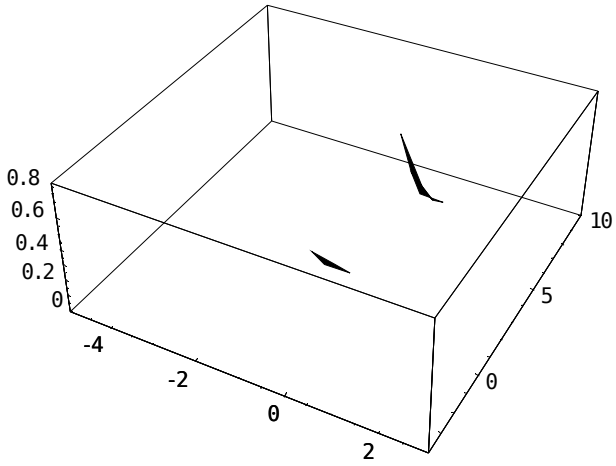
A positive and a Negative pheromone signal combine



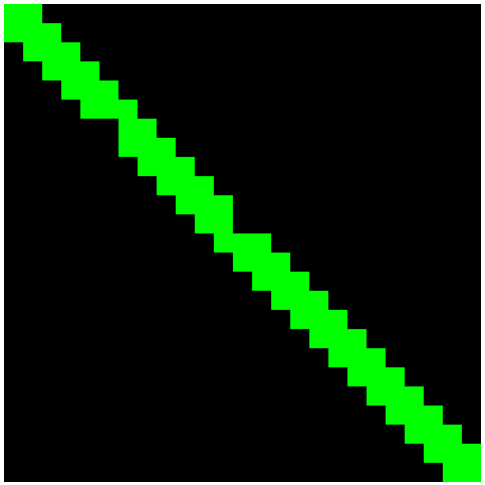
To cancel each other

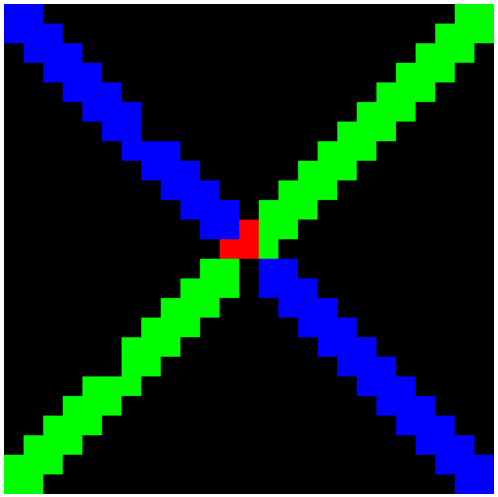


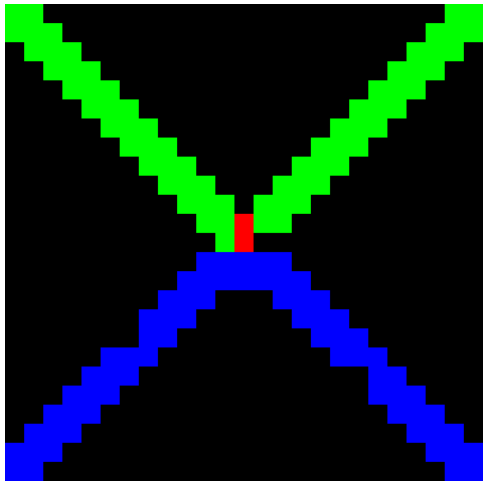


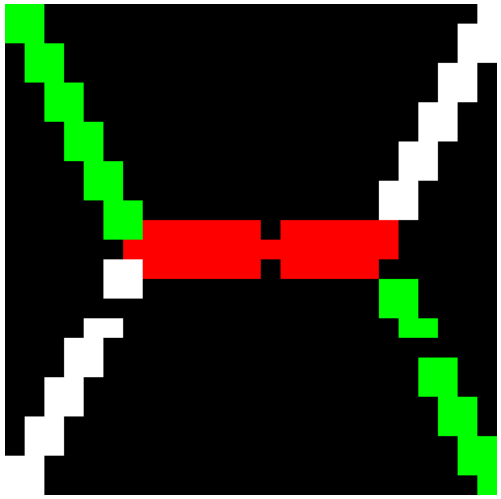


Pheromone results

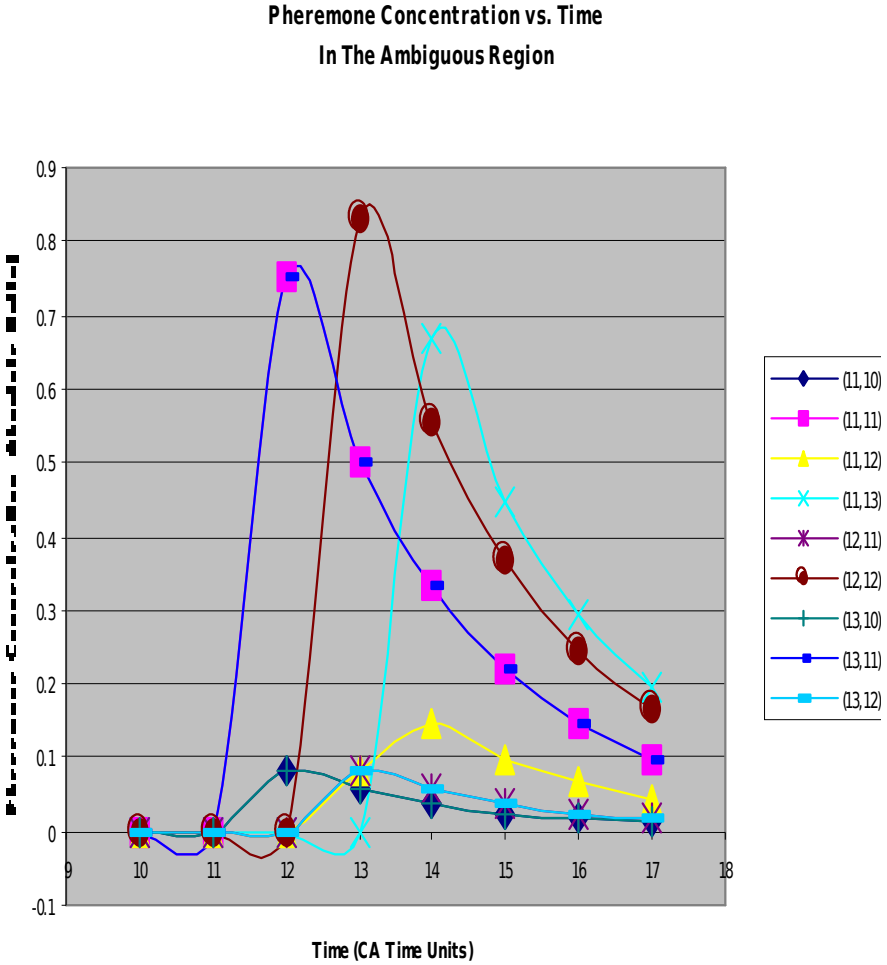


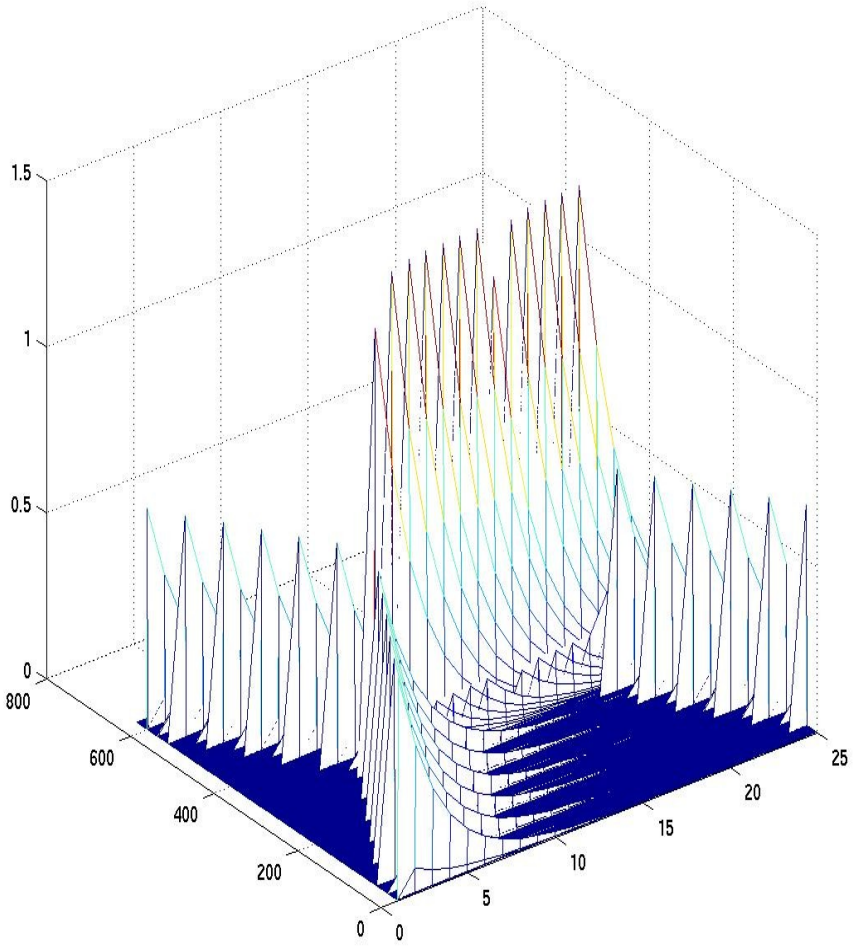






Pheromone concentration





Algorithm 2- EKF

State: Velocity and heading in x and y dimensions

Measurement: Last 3 estimates derived from local collaboration

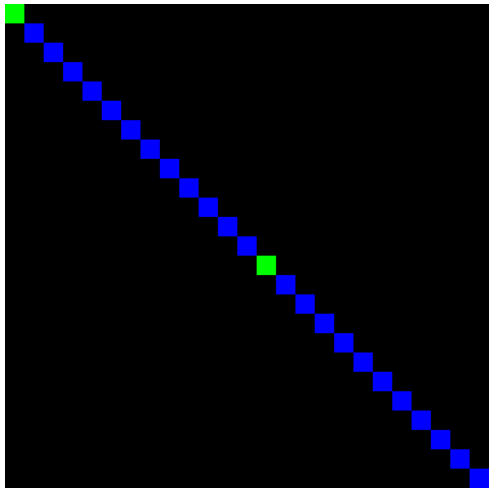
Disambiguation: Compute Euclidean distance between candidate tracks and current estimate. Use minimum.

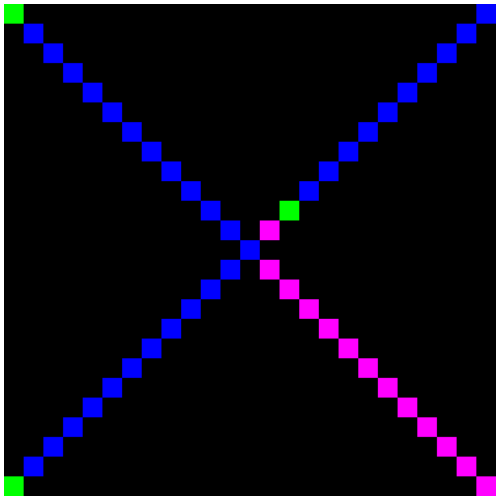
Merge: Kalman Filter equations provide estimates and covariance matrix.

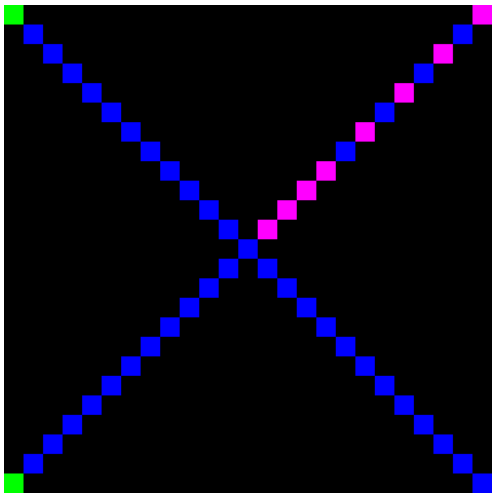
Track initiation: When 3 readings are available estimate is the mean. Covariance matrix is the expected value of the inner product of the estimate minus the readings.

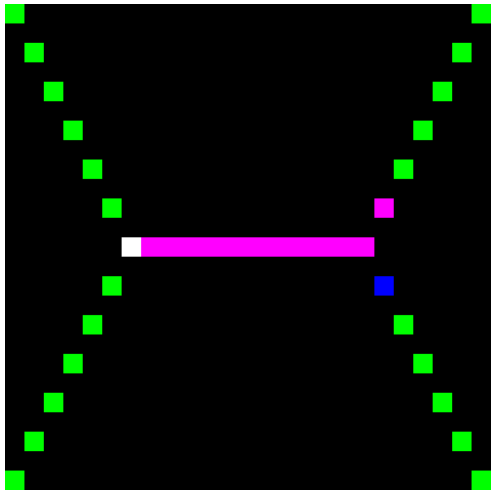
Paper model extended for use in operational demonstration

EKF results

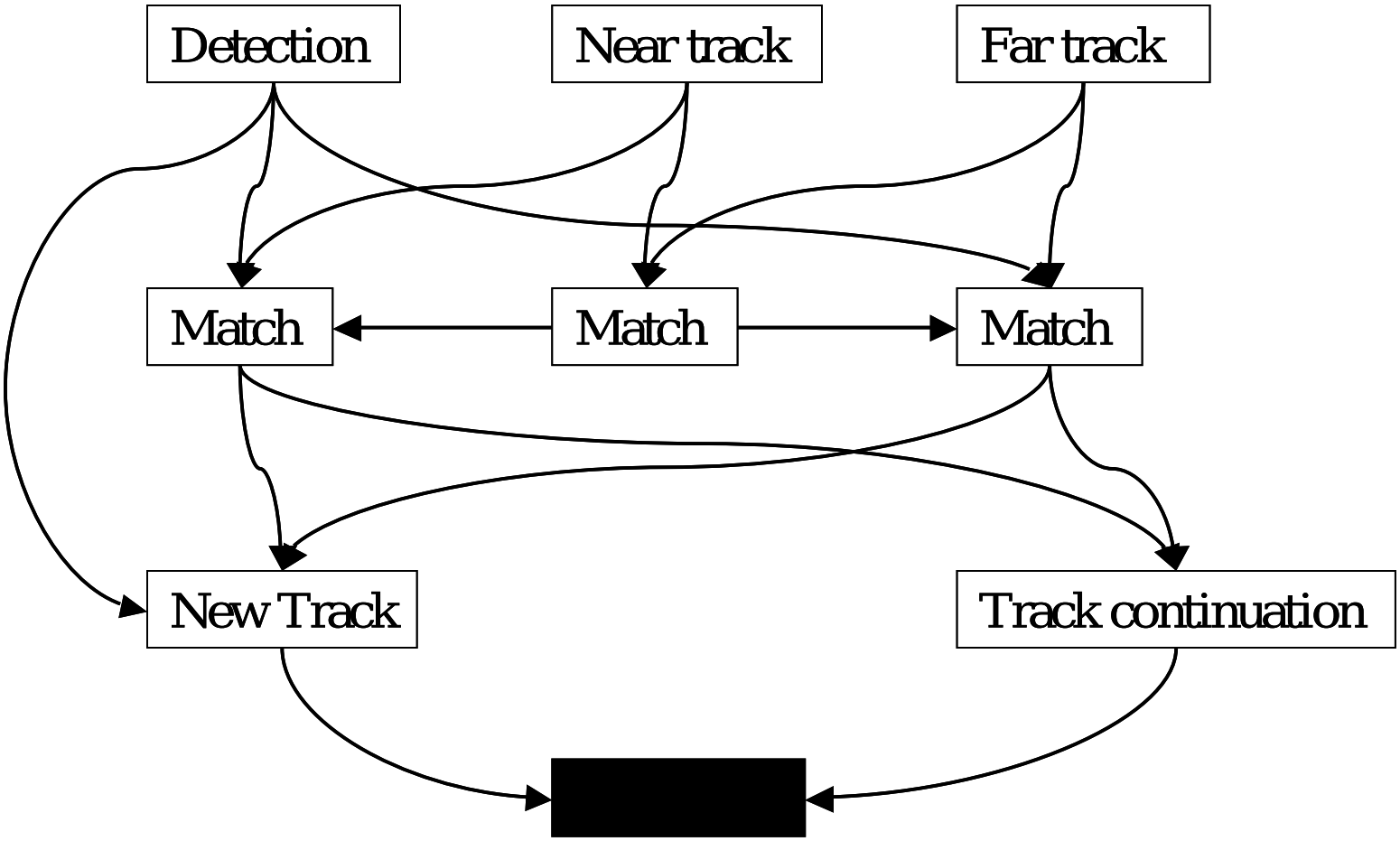




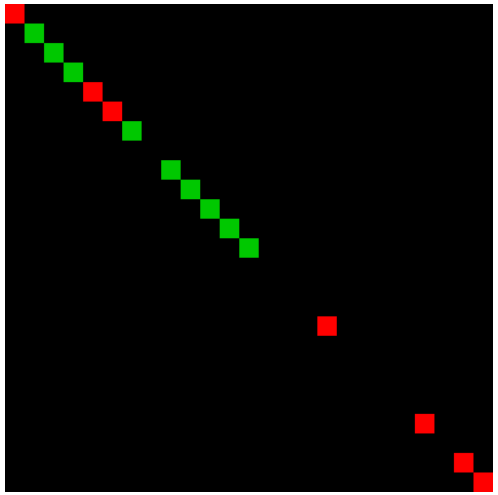


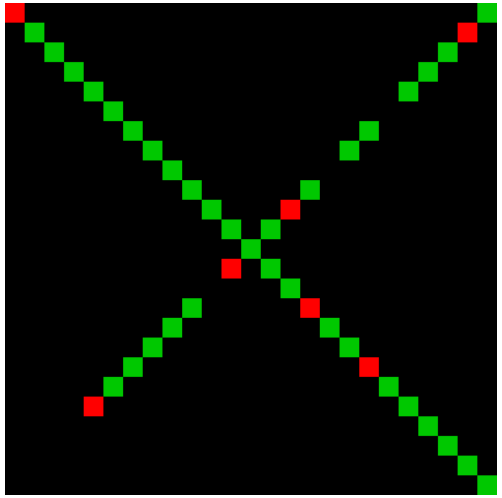


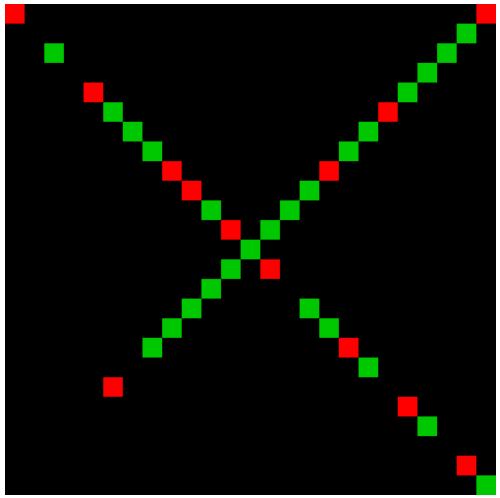
Algorithm 3 – Belief net

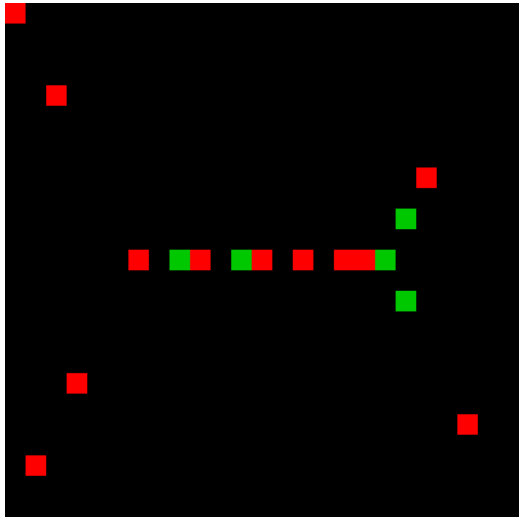


Belief Net results









Conclusion

- Operational demonstration a success
- Makes association tractable by using only local data
- Self-organization increases robustness
- Distributed systems design based on interacting automata
- Derived fully distributed tracking models
- Further live tests need to be made of tracking concepts
- Testing mobile code support for heterogeneous implementations to be done